



**NAVAL  
POSTGRADUATE  
SCHOOL**

**MONTEREY, CALIFORNIA**

**Location Fusion in Land Combat Models**

by

Alan Washburn

December 2004

**Approved for public release; distribution is unlimited.**

Prepared for: TRAC-WSMR, Bldg. 1400  
Martin Luther King Dr.  
White Sands Missile Range, NM 88002-5002

## Report Documentation Page

*Form Approved  
OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE <b>00 DEC 2004</b>	2. REPORT TYPE <b>N/A</b>	3. DATES COVERED <b>-</b>		
4. TITLE AND SUBTITLE <b>Location Fusion in Land Combat Models</b>		5a. CONTRACT NUMBER		
		5b. GRANT NUMBER		
		5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)		5d. PROJECT NUMBER		
		5e. TASK NUMBER		
		5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>Department of Operations Research Naval Postgraduate School Monterey, CA 93943-5000</b>		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)		
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release, distribution unlimited</b>				
13. SUPPLEMENTARY NOTES <b>The original document contains color images.</b>				
14. ABSTRACT				
15. SUBJECT TERMS				
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT <b>UU</b>	18. NUMBER OF PAGES <b>11</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>			

**NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CA 93943-5001**

RDML Patrick W. Dunne, USN  
President

Richard Elster  
Provost

This report was prepared for and funded by TRAC-WSMR, Bldg. 1400, Martin Luther King Dr., White Sands Missile Range, NM 88002-5002.

Reproduction of all or part of this report is authorized.

This report was prepared by:

---

**ALAN WASHBURN**  
Professor of Operations Research

Reviewed by:

---

**LYN R. WHITAKER**  
Associate Chairman for Research  
Department of Operations Research

Released by:

---

**JAMES N. EAGLE**  
Chairman  
Department of Operations Research

---

**LEONARD A. FERRARI, Ph.D.**  
Associate Provost and Dean of Research

# Location Fusion in Land Combat Models

by

Alan Washburn

## **ABSTRACT**

This document discusses methods for representing and fusing location information in combat models, distinguishing between Monte Carlo and analytic models. The Kalman filter is emphasized as a practical and accurate method for fusing new and old information. One particularly useful Kalman filter, the Maneuvering Target Statistical Tracker (MTST), is dealt with explicitly. Fused information is often represented graphically by ellipses, the equivalent of a Kalman filter's covariance matrix. The connection between elliptical dimensions and kill probabilities is also reviewed.

## **1. Introduction**

Any combat model needs to represent information fusion, whether implicitly or explicitly. The extreme cases of perfect ignorance or perfect knowledge are usually made implicitly. This is the case, for example, in systems of Lanchester equations, or in stochastic duels. However, information is sufficiently important to the Future Combat System that combat models that claim to represent FCS must deal explicitly with problems of imperfect information acquisition and fusion. Hence the Army's recent emphasis on realistic models of fusion within AWARS (an analytic model) and CASTFOREM (a Monte Carlo simulation). This document deals mainly with one particularly important component of fusion: the fusion of information about entity locations. The Monte Carlo paradigm is particularly apt fusion-wise on account of its explicit representation of quantities such as measurements and true entity locations. Fusion within analytic models is also possible, but more difficult.

## **2. Fusion—What Is It?**

Fusion can be done at multiple levels according to the Joint Director of Laboratories (Waltz and Llinas, 1990). This report is about level 1, the level having to do with position and identity estimates. We suppose that here is a collection of independent “entities” (squads, tanks, command posts,...), each of which has a state that is unknown and may change unpredictably with time. Measurements are made on some irregular schedule, possibly in batches, and fusion is the process of updating our knowledge of the state of each entity. It is assumed that each measurement is caused by a single entity. The process of figuring out which entity caused each measurement is data association. Once the data is associated, the process of incorporating it into the state estimate for the appropriate entity is filtering. Both of these real world processes need to be imitated, to some degree, in combat models. Although data association must logically precede filtering, the two parts will be discussed in reverse order.

### 3. Filtering of Position

An entity's state vector may incorporate a diversity of properties, but the properties nearly always include position information. Position information can be updated using a Kalman filter. Kalman filtering is a widely accepted technique with over four decades of development and experience behind it, and has the further benefit of employing only simple mathematics, so it is natural to use it in combat models. The basic idea in Kalman filtering is to fuse the most recent measurement with the previous state estimate to produce an updated state estimate. In all cases the state estimate takes the form of the mean ( $\mu$ ) and covariance matrix ( $\Sigma$ ) of the distribution of the true state of the entity. The weight put on the previous state estimate should depend on how old it is, as well as on  $\Sigma$ , since the entity may have moved in the meantime. A Kalman filter thus needs two parts: a motion model and a measurement model.

**Motion model MTST.** The Navy makes frequent use of a motion model called MTST (Maneuvering Target Statistical Tracker). MTST applies to targets moving freely in a two-dimensional, frictional medium. The MTST state vector is  $[X, Y, U, V]'$ , where  $(X, Y)$  is the entity's position and  $(U, V)$  is its velocity. Here the ' indicates transpose, since the state vector is represented as a column vector.

Each velocity component is assumed to be an Ornstein-Uhlenbeck (OU) process, the simplest normal, stationary process that fluctuates around 0. An OU process has two parameters: the relaxation time  $\tau$  and the root-mean-square velocity  $s$ . The average velocity is 0 in an OU process, befitting targets that move in a frictional medium such as land or sea, but the root-mean-square velocity is not. The relaxation time  $\tau$  is the time that it usually takes for the velocity to change significantly, perhaps an hour for a submarine, 0.2 hours for a tank, or 4 hours for a regiment. These two parameters ( $\tau$  for time and  $s$  for velocity) are the minimum number required to provide scales for both time and velocity. If  $U(t)$  is the (say E-W) velocity at time  $t$ , then the OU motion model over a time period of length  $\Delta$  is  $U(t + \Delta) = \rho U(t) + W$ , where  $\rho = \exp(-\Delta/\tau)$  and  $W$  is a normal "process noise" with a variance chosen to make the variance of  $U(t)$  be  $s^2$  at all times, regardless of  $\Delta$ . That variance is  $q = s^2(1 - \rho^2)$  as can be readily verified using the formula for the variance of a sum of independent random variables. The process noise  $W$  represents the essential unpredictability of the entity's motion. The entity's location  $X(t)$  is assumed to be the time integral of its velocity. The state update for motion in a Kalman filter takes the form  $\mu(t + \Delta) = \Phi \mu(t)$ , and  $\Sigma(t + \Delta) = \Phi \Sigma(t) \Phi' + Q$ . For MTST,  $\Phi$  and  $Q$  are the following  $4 \times 4$  matrices:

$$\Phi = \begin{bmatrix} 1 & \tau(1 - \rho) & & \\ & 1 & \tau(1 - \rho) & \\ & & \rho & \tau(1 - \rho) \\ & & & \rho \end{bmatrix} \text{ and } Q = \begin{bmatrix} \varepsilon_1 & & \varepsilon_2 & \\ & \varepsilon_1 & & \\ & & s^2(1 - \rho^2)/2 & \\ & & & s^2(1 - \rho^2)/2 \end{bmatrix}$$

The two second-order terms can safely be taken to be 0 when  $\Delta$  is small compared to  $\tau$ , but for completeness, the exact expressions are

$$\varepsilon_1 = \tau s^2(2\Delta - \tau(3 - 4\rho + \rho^2)) \text{ and } \varepsilon_2 = \tau s^2(1 - \rho)^2.$$

Points worthy of note about MTST are that:

- The motion update algebra is simple to do on a computer, amounting to one exponentiation, a few additions, and a few multiplications. The motion update is generally done only as part of the preparation for processing a measurement, but could also be done at the behest of some person/subroutine/agent interested in the current state of the entity.
- When the time since the last measurement is small,  $\tau(1-\rho)$  is approximately  $\Delta$ , in which case the position updates for X and Y take the familiar form of adding the previous velocity estimate multiplied by  $\Delta$ .
- When the time since the last measurement is large,  $\rho$  is approximately 0, and the position updates multiply previous velocity estimates by no more than  $\tau$ . Without any intervening measurements, the location estimate moves in the direction of the most recent velocity estimate, but eventually comes to a stop as the corresponding velocity estimate approaches 0. This is in accordance with the OU process for velocity—regardless of the current velocity, the best estimate a long time in the future (for an entity subject to friction) is 0. The phenomenon has nonetheless caused some consternation among Navy officers, who initially expect the target to move indefinitely at the most recently estimated course and speed. The problem disappears if there are intervening measurements.
- The two parameters cover stationary targets ( $s=0$ ,  $\tau$  anything), high speed targets like helicopters ( $s=50$  kt,  $\tau=0.1$  hour), and medium speed entities like tanks ( $s=5$  kt,  $\tau=.2$  hour). A separate logic is not needed for every kind of entity. There would need to be a database providing  $\tau$  and  $s$  for the various entity types. Since it is not essential to get exactly the right movement model, it might be possible to use the same default parameters for a large class of entities (“tracked”, say).
- The MTST code in VBA or MATLAB is available from the author.

**Measurement.** One of the advantages of a Kalman filter is that there does not have to be a single kind of measurement. The only important feature is that every measurement be of some quantity or quantities corrupted by additive, normally distributed noise. Possibilities include:

- A two-dimensional measurement of  $(X, Y)$  made by a UAV.
- A one-dimensional measurement of  $\arctan(Y/X)$ , the bearing to the target, made by an ESM intercept. This one has been much explored by the Navy.
- A one-dimensional measurement of  $(U^2 + V^2)^{0.5}$ , the entity’s speed, made by a seismic sensor.
- A three-dimensional measurement consisting of range, bearing, and Doppler shift, made by a radar.

All such measurements can be used to update  $\mu$  and  $\Sigma$ , thus fusing the measurements with what was known before, properly weighted for age. This all happens automatically in a Kalman filter, and the mathematics is efficient, once correctly implemented. Details can be found in many textbooks, or in my notes at <http://diana.cs.nps.navy.mil/~washburn/>.

## 4. Data Association

A measurement may contain clues or signatures that uniquely determine the target that caused it. There is no data association problem in that case, and fusion amounts to updating the state of whatever entity was the cause. A data association problem emerges only when multiple entities might reasonably have caused the measurement. Most data association schemes assume that the measurement to be processed was caused by the entity that is in some sense closest to it. In the case of the Kalman filter, the best distance to use theoretically is the Mahalanobis distance, or, as I call it, the dimensionless shock DS. If there are too many possibilities to permit the calculation of DS for each one, it may be attractive to eliminate most of them through some sort of a gating scheme.

Regardless of the association scheme, it is inevitable that an association mistake will eventually be made. The mistake might be to associate the measurement with the wrong target, or to fail to initiate a new target track when the cause of the measurement is in fact a new target, or to fail to simply ignore the measurement when the cause is not related to the military problem (a false alarm). Such mistakes can have very serious consequences in the real world—one aircraft accidentally shot down in OIF was the victim of an association mistake. The problem is to imitate this in a combat model. The method of doing so must depend on whether the model is a Monte Carlo simulation like CASTFOREM, or an analytic model like AWARS.

## 5. Monte Carlo Fusion

One of the beauties of dealing with a Monte Carlo simulation is that the true entity causing any particular measurement is known to the analyst. Therefore one association option is to adopt the policy of always associating data correctly, the “Truth” policy. There is much to be said for it. Any model of combat is going to miss real world aspects that are important association cues, and will therefore do a worse job of association than happens in the real world. However, the real world can certainly do no better than Truth, so Truth constitutes an upper bound on how well any association scheme can do. There are additional arguments that favor retaining Truth as one of the association options. The creation and destruction of false target tracks would not be necessary, for example.

Using Truth does not imply that the perception of an entity’s state will always be perfect. That perception depends on the frequency and quality of measurements, which are computed as usual. For entities tracked by a Kalman filter, it is worthwhile to compute DS for each measurement, as is currently done in CASTFOREM. That distance is worth keeping track of even when Truth is employed. Statistics about DS will always be interesting in the sense that large values correspond to mistakes that might have been made, except for the employment of Truth.

In addition to Truth, there should perhaps be some kind of an automatic association scheme. The simplest such scheme would irrevocably associate each measurement with some entity, with no possibility of subsequent revision. Call this the Single Hypothesis Tracker, or SHT. When the minimized DS is sufficiently small (smaller than 10, say), the measurement would be associated with the corresponding entity. If it is larger than 10, it might simply be ignored on the grounds that, in the real world, it would probably be a false alarm. In a scenario where entities might plausibly be created or discovered as time goes by, a large DS might also cause the creation of a new entity. The SHT will occasionally make mistakes, since DS is occasionally small for a false target, or large for

a true one.

The SHT is surely a simpler and less effective approach to data association than the one proposed by Orincon for inclusion in the FCS. As such, it constitutes a lower bound on the effectiveness of the FCS system, just as Truth constitutes an upper bound. There is thus a good argument for including both Truth and SHT, especially if the actual FCS system is too complicated to retain in a simulation.

There are also trackers that remember multiple hypotheses about which data association is correct, sometimes (as in Stone et al., 1999) with explicit probabilities for the correctness of the hypotheses. I do not see the argument for including them in an existing combat model unless the *purpose* of running the model is to be examination of the effectiveness of various fusion schemes.

It should be recognized that the SHT, or any multiple hypothesis extension, will be of no use for measurements that do not involve position. Kalman filters work only for real-valued state components that are plausibly normally distributed.

## **6. Analytic Fusion**

An analytic model such as AWARS (Glasgow and Bailey, 2004) will have more difficulty including association phenomena than will a Monte Carlo simulation. The true values of state variables are easily handled in Monte Carlo simulations by sampling the appropriate distributions. In an analytic model, the only really correct equivalent is a numeric integral over all possible values of the state variables, a much more difficult operation. Replacing distributions by their means (expected-value analysis, or EVA) may be the only practical alternative. In fact, EVA\_Truth is the modal method for representing fusion in analytic models; the association is always correct, and actual measurements are not represented. It is thus almost inevitable that an analytic model will provide optimistic answers to problems where fusion is a significant part of the problem. The developers of AWARS recognize this (Glasgow and Bailey, 2004, page 2 and footnote 5). It is not practical to implement something like the SHT in an analytic model, so there is no lower bound counterbalance EVA\_Truth.

There are nonetheless some powerful reasons for employing analytic models, so fusion needs to be represented in spite of the difficulties.

Kalman filtering can still be employed in analytic models, even without specific measurements being available. The best opportunity is with location information, as in Monte Carlo models, and MTST remains an attractive candidate for the same reasons. A Kalman filter will automatically deal with issues of data aging, can handle measurements of multiple kinds (bearings, in particular), and will automatically provide the kind of information that is required to compute kill probabilities. The state estimates themselves are in some cases trivial because of the EVA assumption, but the covariance matrix computations are still realistic and useful. In fact, it is a curious characteristic of Kalman filters that the covariance computations are independent of the actual measurements, and therefore of the EVA assumption.

In two dimensions, the covariance matrix determines the size and orientation of the ellipse that graphically represents the target's location. Since computations of kill probabilities rely on knowing the dimensions of that ellipse, there is a good argument for computing its dimensions as accurately as possible.

## 7. Kill Probabilities

There is no point in fusing information unless the information will be used to support some decision. Decision making in the real world will be complex, and will involve targeting decisions about *what* should be shot *when* at *whom*. Fusion levels far in excess of level 1 will be involved, but level 1 is still important because it influences the effect of fire. This section deals with heuristics for allocating fire and assessing its effectiveness, based on the accuracy of target location.

**Notation.** The models below differ in how lethality is modeled, but all share the same error structure. Target location error and weapon dispersion error (s) are each assumed to be unbiased and elliptically normally distributed, and all errors are assumed to be independent. Target location errors are assumed to be fused errors from the tracking model, evaluated at the time of fire. Let

$\sigma_x$ =horizontal dispersion standard deviation;

$\sigma_y$ =vertical dispersion standard deviation;

$\sigma_u$ =horizontal target location standard deviation; and

$\sigma_v$ =vertical target location standard deviation.

Note that  $(U, V)$  is now the target location error, rather than an entity's velocity. Two separate two-dimensional errors are involved because the target location errors are assumed to be common to all shots at a target, whereas the dispersion errors of different shots at the same target are all independent. The assumption is that all shots are taken "simultaneously" in the sense that no location information is fused in the time between shots.

**Carleton weapons.** In this case,  $P_k = \exp(-R^2/LR^2)$ , where  $R$  is the Euclidean distance between the target and the impact point of the weapon, and  $LR$  is data. This formula is also known as the "diffuse Gaussian" formula. It is to be distinguished from the cookie-cutter or "definite range" formula where damage occurs if and only if  $R$  is smaller than  $LR$ . The Carleton expression generally results in significantly simpler expressions than the cookie-cutter formula, especially when multiple shots are made at the target (Washburn, 2003). For a single shot aimed at the center of the target distribution,

$$E(P_k) = LR^2 / \{(LR^2 + 2(\sigma_x^2 + \sigma_u^2))^{1/2} (LR^2 + 2(\sigma_y^2 + \sigma_v^2))^{1/2}\}.$$

The expected value operator  $E()$  would normally not be written. This is formula 2-9 from (Washburn, 2004), except that  $LR^2 = 2b^2$  and  $\mu_x = \mu_y = 0$  in that document; i.e., the weapon is aimed directly at the target. See also chapter 2 of Eckler and Burr (1972). In a Monte Carlo model with imperfect correlation, it is conceivable that one might not be aiming at the center of the target distribution, either because of a correlation mistake or because the target in question is not the one aimed at. Whether the model is Monte Carlo or analytic, the general form of formula 2-9 still applies and is simple.

**Cookie-cutter weapons.** In this case there is a simple formula only when the total horizontal and vertical errors have the same standard deviation, say

$\sigma^2 \equiv \sigma_x^2 + \sigma_u^2 = \sigma_y^2 + \sigma_v^2$ . The formula is

$$P_k = 1 - \exp\left(-\frac{LR^2}{2\sigma^2}\right).$$

$LR$  might be the lethal radius of a weapon, or it might be the radius of a circular pattern of weapons, in which case  $Pk$  is merely the probability that the target is included within the pattern. For the ICM weapon, this would need to be multiplied by the probability that the target is killed, given that it is within the pattern.

If the horizontal and vertical errors are not equal, there is no simple expression for  $Pk$  for cookie-cutter weapons.

**Patterns.** When target location errors are large, a single round will not be effective. In that case it may be wise to use a pattern of several rounds, but the subject is complicated by the lack of independence between rounds (recall that they all share the same target location error). The weapons should generally be aimed in a pattern, which raises the subject of pattern optimization. No combat model wants to get into this subject in detail, but a simple formula relating the inputs to the optimized kill probability may still be useful. I suggest considering the “square-root formula”, which is formula (3-10) in Washburn (2002). The only important quantity is

$$z = na/(2\pi\sigma_U\sigma_V),$$

where  $n$  is the number of rounds/bomblets and  $a$  is the lethal area per round. The dispersion errors are unimportant. The formula is

$$Pk = 1 - (1 + \sqrt{2z}) \exp(-\sqrt{2z}).$$

If the pattern can only be optimized to the extent that all bomblets will be uniformly distributed over a circle of the shooter’s choice, then a slightly smaller kill probability results, namely

$$Pk = (1 - \exp(-\sqrt{z}))^2.$$

Either of these formulas is simple enough to be evaluated many times in the process of deciding how to shoot at a target, possibly by seeking the method that is most cost-effective in the sense of maximizing  $Pk$  per dollar. Either is sufficiently accurate if the number of bomblets is large (say, more than 10).

If exact pattern answers are required in the Carleton case for 10 or fewer bomblets, see Washburn (2003). The accompanying spreadsheet includes VBA code that will evaluate  $Pk$  for an optimal pattern, as long as the number of weapons involved is not large.

## REFERENCES

Blackman, S. 1986. *Multiple-Target Tracking with Radar Applications*, Artech House, Boston.

Colgrove, S. 2004. “Notes on CASTFOREM fusion”, AMSAA.

Eckler, R. and Burr, S. 1972. *Mathematical Models of Target Coverage and Missile Allocation*, MORS.

Glasgow, S. and Bailey, T. 2004. “Intelligence Fusion Process for the Advanced Warfighting Simulation (AWARS)”, TRAC-TD-3-004, TRAC-FLVN.

Stone, L., Barlow, C., and Corwin, T. 1999. *Bayesian Multiple Target Tracking*, Artech House, Boston.

Waltz, E. and Llinas, J. 1990. *Multisensor Data Fusion*, Artech House, Boston.

Washburn, A. 2003. “Diffuse-Gaussian multiple-shot patterns”, *Military Operations Research* Vol. 8, No. 3.

Washburn, A. 2002. “Notes on Firing Theory”, available 11/04 at <http://diana.cs.nps.navy.mil/~washburn/>.

Washburn, A. 2004. “A Short Introduction to Kalman Filters”, available 11/04 at <http://diana.cs.nps.navy.mil/~washburn/>.

## **INITIAL DISTRIBUTION LIST**

1. Research Office (Code 09).....1  
Naval Postgraduate School  
Monterey, CA 93943-5000
2. Dudley Knox Library (Code 013).....2  
Naval Postgraduate School  
Monterey, CA 93943-5002
3. Defense Technical Information Center.....2  
8725 John J. Kingman Rd., STE 0944  
Ft. Belvoir, VA 22060-6218
4. Richard Mastowski (Editorial Assistant).....2  
Department of Operations Research  
Naval Postgraduate School  
Monterey, CA 93943-5000
5. Professor Alan Washburn .....1  
Department of Operations Research  
Naval Postgraduate School  
Monterey, CA 93943-5000
6. Mr. David Kelly .....2  
TRAC-WSMR, Bldg. 1400  
Martin Luther King Dr.  
WSMR, NM 88002-5002
7. Captain Aaron Van Alstine.....2  
TRAC-MTRY  
Naval Postgraduate School  
Monterey, CA 93943-0692